Operator Dependent Interpolation in Algebraic Multigrid

A. Krechel
K. Stüben

German National Research Center for Information Technology (GMD)
Institute for Algorithms and Scientific Computing (SCAI)
Schloss Birlinghoven, D-53754 St. Augustin, Germany

e-mail: krechel@gmd.de, stueben@gmd.de

November 10, 1999

Arbeitspapier der GMD 1048, January 1997

This paper will appear in the Proceedings of the Fifth European Multigrid Conference, EMG'96, held in Stuttgart, Oct 1-4, 1996: Lecture Notes in Computational Science and Engineering 3, Springer, 1998

Abstract. Algebraic multigrid (AMG) methods are based on algebraically defined multigrid components of which, in particular, a proper definition of interpolation is important for obtaining fast and robust convergence. This is because AMG convergence crucially depends on how well the range of interpolation approximates the range of the smoothing operator used. On the basis of various experiments, we will demonstrate the dependency of convergence on the interpolation operator. A simple improvement by means of a Jacobi relaxation step, applied to the interpolation, is shown to considerably enhance convergence and robustness. Relaxation of interpolation can also be used to improve the performance of algebraic multigrid approaches which are based on accelerated coarsening strategies. Finally, in a parallel environment, the use of local relaxation of interpolation (only along processor boundaries) may be used to stabilize convergence.

1 Introduction

In any multigrid approach, smoothing (by some kind of relaxation) and coarse-grid correction are used in conjunction to eliminate the error. This requires that error components which cannot be corrected by appealing to a coarser-grid problem, must be effectively reduced by smoothing. In standard geometric multigrid methods, a fixed grid hierarchy and linear interpolation are used, so that relaxation must be chosen which smooths the error in the usual geometric sense. More robust approaches employ sophisticated multigrid components such as complex smoothers, operator-dependent interpolation, Galerkin operators and/or multiple semi-coarsened grids. The most radical attempt to obtain robustness is the algebraic multigrid (AMG) approach. Here, the complete coarsening process itself, performed automatically and in a purely algebraic setting, is a substantial part of the algorithm.

The first attempt to develop a fairly general algebraic multigrid program, AMG1R5, took place over 10 years ago, initiated by an idea of A. Brandt [5], and investigated in detail in [18], [20], [19]. This program has been developed to solve sparse matrix equations

$$Au = f \quad \text{or} \quad \sum_{j=1}^{n} a_{ij} u_j = f_i \quad (i = 1, 2, ..., n) \tag{1}$$

with primarily *positive definite*, *off-diagonally non-positive* matrices A in mind (M-matrices). The central idea was to employ plain Gauss-Seidel "point" relaxation for smoothing and automatically construct all coarser levels and interpolation operators so that the range of interpolation is *forced* to (approximately) contain those "functions" which are unaffected by relaxation. This automatism gives AMG a very high flexibility in adapting itself to specific requirements of the problem to be solved and makes it applicable to various problems which are out of the reach of usual geometric multigrid.

If applied to scalar second order elliptic PDEs, AMG's flexibility in adapting the coarsening *locally* to, for instance, varying anisotropies, is the reason why AMG can employ simple pointwise relaxation for smoothing without the need for multiple semi-coarsened grids. Moreover, since AMG operates merely on the given algebraic equations (1), it can directly be applied to 3D problems as well as to problems on unstructured grids. AMG1R5 has also been tested for certain discretized PDE problems not satisfying the above assumptions (e.g., involving non-symmetric matrices and matrices containing some "small" positive off-diagonal entries) where it still worked but convergence sometimes slowed down. A potential reason for this slow-down is that the interpolation used in AMG1R5 is not necessarily appropriate in such situations: although no geometric information has explicitly been exploited, the definition of interpolation was still motivated by geometric arguments.

The purpose of this paper is to improve interpolation and to investigate its influence on the overall convergence. Essentially, a simple Jacobi relaxation step, applied to the interpolation operator, is used to force the ranges of relaxation and interpolation to be closer together. Although there is much room for generalizations, in this basic study, our focus is on scalar PDE problems. We will show that the AMG performance for some problems which can be regarded as being difficult even for robust geometric multigrid, will increase substantially. For other cases such as discrete Poisson-like equations, no substantial benefit is obtained, at least not if numerical work is taken into account. Besides

increasing robustness in general, there are other particular situations where relaxation of interpolation may lead to substantial enhancements. We will present preliminary results refering to aspects like *accelerated coarsening* and *parallel coarsening* (see Section 4.5).

During the last years, there was a noticeable increase of research on algebraically oriented multigrid methods, for example, see [1], [2], [3], [6], [7], [9], [10], [11], [12], [14], [15], [17], [21], [22] and others. Some of these papers also dealt with improvements of interpolation. For instance, Jacobi relaxation has recently been used for obtaining reasonable interpolation in the context of a different algebraic multigrid approach (based on "aggregation of elements" [21], [6]). In [7], a modification of the original AMG1R5 interpolation has been proposed which takes positive off-diagonal entries of A into account (which have simply been ignored in AMG1R5). While this modification is still based on geometric assumptions (in particular, the size of matrix entries is assumed to reflect the distance between grid points), relaxation of interpolation leads to improvements in a simple and purely algebraic way.

We want to point out that the current solver is at an experimental stage and far from being optimized. This refers mainly to the startup phase but, to some extent, also to the solution phase. Thus, it does not yet allow to draw final conclusions regarding the overall efficiency achievable at the end.

2 Motivation of AMG

Formally, an AMG cycle can be described in the same way as a geometric multigrid cycle, except that terms like *grid, subgrids, grid points,* etc. have to be replaced by *set of variables, subsets of variables, single variables,* etc. Since the formal extension of a two-level method to a real multi-level method is straightforward, we here consider only the two-level method with indices h and H distinguishing the levels. In particular, we re-write (1) as

$$A_h u^h = f^h \quad \text{or} \quad \sum_{j \in \Omega_h} a_{ij}^h u_j^h = f_i^h \quad (i \in \Omega_h) \tag{2}$$

with Ω_h denoting the index set $\{1, 2, ..., n\}$. From this fine-level system, AMG automatically constructs a coarse-level system

$$A_H u^H = f^H \quad \text{or} \quad \sum_{l \in \Omega_H} a_{kl}^H u_l^H = f_k^H \quad (k \in \Omega_H \subset \Omega_h) \tag{3}$$

which formally plays the same role as coarse-grid equations in normal multigrid methods. In particular, f^H and u^H will be *residuals* and *corrections*, respectively. (In the sequel, we usually denote solution and correction quantities by the letters u and e, respectively.) The assumption $\Omega_H \subset \Omega_h$ in (3) means that we regard the set of coarse-level variables as a subset of the fine-level ones. That is, the coarse-level variable u_k^H is used to directly correct its corresponding fine-level analog, u_k^h. Note that this is different from algebraic multigrid approaches based on "aggregation of elements" (see, e.g., [21], [3], [6], [12]).

2.1 Formal AMG Components

We assume Ω_h to be split into two disjoint subsets

$$\Omega_h = C_h \cup F_h \quad \text{and} \quad \Omega_H := C_h \tag{4}$$

with C_h representing those variables which are also contained in the coarse level *(C-variables)* and F_h being the complementary set *(F-variables)*. We formally assume vectors and matrices to be ordered according to this C/F-splitting and we write $u^h = (u_F, u_C)^T$ and use B_{FC} to denote an operator (matrix) which maps vectors of C-variables into vectors of F-variables. Analogously, we write B_{CF}, B_{FF} and B_{CC}.

We generally assume the transfer operators, I_H^h *(interpolation)* and I_h^H *(restriction)*, to be of the following (full rank) forms

$$I_H^h = \begin{pmatrix} I_{FC} \\ I_{CC} \end{pmatrix}, \quad I_h^H = (I_{CF}, I_{CC}) \tag{5}$$

with $I_{CC} := \mathbb{I}_{CC}$ (identity operator). Coarse-level operators are defined via the *Galerkin operator*

$$A_H := I_h^H \, A_h \, I_H^h \, . \tag{6}$$

Finally, we need a smoothing process with a corresponding *smoothing operator* G_h (linear iteration operator). In AMG, the standard smoothing process is plain *pointwise Gauss-Seidel relaxation* (in some order, normally C/F).

Once all the above components are known, an iterative two-level method can be set up as in normal geometric multigrid with the well-known *coarse-level correction operator* $K_{h,H}$ and *two-level iteration operator* $M_{h,H}$:

$$M_{h,H} = G_h^{\nu_2} \, K_{h,H} \, G_h^{\nu_1} \quad \text{with} \quad K_{h,H} := (\mathbb{I}_h - I_H^h A_H^{-1} I_h^H A_h) \, . \tag{7}$$

Summarising, what needs to be defined in order to formally set up a two-level (and recursively a multi-level) process, are just the C/F-splitting and the operators I_{FC} and I_{CF}.

2.2 Direct Solver Aspects

The original equations (2), re-written in block form

$$\begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} u_F \\ u_C \end{pmatrix} = \begin{pmatrix} f_F \\ f_C \end{pmatrix}, \tag{8}$$

have been used as a starting point for the development of algebraically oriented two-level methods in various papers (e.g. [1], [2], [8], [9], [15]). Such methods can be regarded as approximations to the block Gauss-elimination procedure applied to (8) with the Galerkin operator (6) being an approximation to the Schur complement, $A_H \approx A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$. The original AMG, although motivated differently in [19], can also be interpreted in this context.

For a motivation, we first recall situations for which the algebraic two-level method degenerates to a *direct solver*, i.e. for which either $K_{h,H} \, G_h = 0$ or $G_h \, K_{h,H} = 0$. For

example, we obviously have $K_{h,H} G_h = 0$ if A_h^{-1} and A_H^{-1} exist and if the "smoothing operator" G_h satisfies[1]

$$G_h = \left(0 \mid I_H^h\right) = \begin{pmatrix} 0 & I_{FC} \\ 0 & I_{CC} \end{pmatrix} . \tag{9}$$

This is well-known and an immediate consequence of the trivial identity $K_{h,H} I_H^h = 0$. If we define the interpolation (5) by solving the equations

$$A_{FF} e_F + A_{FC} e_C = 0 \tag{10}$$

for e_F, that is, $I_{FC} = -A_{FF}^{-1} A_{FC}$, then (9) means

$$G_h = \begin{pmatrix} 0 & -A_{FF}^{-1} A_{FC} \\ 0 & I_{CC} \end{pmatrix} . \tag{11}$$

This "smoothing operator", in turn, corresponds to exactly solving the F-equations for u_F, i.e.

$$A_{FF} u_F + A_{FC} u_C = f_F \tag{12}$$

or, in terms of the error,

$$e_F = -A_{FF}^{-1} A_{FC} e_C . \tag{13}$$

For this particular choice of I_{FC} it follows – independently of I_{CF} – that the Galerkin operator (6) is just the Schur complement corresponding to (8) and, as such, is regular if both A_h and A_{FF} are regular. This result and an analogous one involving "post-smoothing" are summarized in the following Lemma.

Lemma: *Let A_h and A_{FF} be regular and let G_h be defined by (11). Then the following is true:*

1. *If $I_{FC} = -A_{FF}^{-1} A_{FC}$ then A_H^{-1} exists and $K_{h,H} G_h = 0$.*
2. *If $I_{CF} = -A_{CF} A_{FF}^{-1}$ then A_H^{-1} exists and $G_h K_{h,H} = 0$.*
3. *In either of the two cases, the Galerkin operator (6) is just the Schur complement corresponding to (8), i.e. $A_H = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$.*

Concerning the second statement in the lemma, we first observe that $I_h^H A_h e^h = A_H e_C$ holds for all $e^h = (e_F, e_C)$. From this and the trivial identity $I_h^H A_h K_{h,H} = 0$ it immediately follows that the application of the coarse-level-correction operator $K_{h,H}$ yields error vectors e^h with $e_C = 0$ and therefore, due to (13), post-smoothing reduces the total error to 0.

It follows from the lemma that, in order to obtain a direct method, only one of the transfer operators has to be explicitly specified. For the two-level method to be a direct solver independently of whether pre- or post-smoothing is used, one has to specify both transfer operators accordingly. Note that, for symmetric A^h, we have $I_{CF} = I_{FC}^T$, i.e. the restriction is the transpose of interpolation. For non-symmetric problems, I_{CF} is the transpose of a different interpolation, namely, the one corresponding to A_h^T.

[1] Although this operator is not a practical smoothing operator, we formally stick to the standard multigrid terminology.

2.3 Approximation of the Direct Solver

Except for very particular cases (such as 1D differential problems), the above direct solvers are not practical, they serve only as a guideline for more realistic approaches. Generally, the explicit computation of A_{FF}^{-1} is too expensive and a recursive application in a multilevel context would be prohibitive due to fill-in on coarser levels. The latter is also true if the C/F-splittings are recursively selected such that the matrices A_{FF} become very simple and do allow for a fast inversion. For instance, if A_{FF} is forced to be diagonal on each level, the reduction of unknowns in the coarsening process quickly becomes extremely slow which, again, leads to drastic fill-in on the coarser levels.

Instead, one may try to approximate A_{FF}^{-1} by a simpler matrix. Various ways have been investigated in the literature (see, for instance, [9], [15]). However, for obtaining a reasonable multilevel method, additional crucial requirements are a fast reduction of unknowns (high dimension of A_{FF}) as well as reasonable sparsity of A_H. Clearly, the availability of efficient approximations to A_{FF}^{-1} strongly depends on the selected C/F-splitting. Our goal is to automatically construct (matrix-dependent) splittings, which allow for an efficient approximation of A_{FF}^{-1}.

More specifically, we will construct splittings which tend to make A_{FF} *strongly diagonally dominant*. This makes sense for large classes of problems, in particular those which are in the focus of this paper, namely, elliptic partial differential equations. We then can most efficiently approximate u_F in (12) by plain Gauss-Seidel relaxation, applied only to F-equations (*F-relaxation*). Likewise, we can approximate e_F in (10) by relaxation. (However, since I_{FC} has to be kept as "local" as possible, only schemes like Jacobi relaxation are appropriate.) Formally, the original AMG algorithm [19] fits into this concept except that the approximation of (10) was not done by relaxation but was partly motivated by geometric reasoning (though no geometric information was explicitely exploited; see also Section 3.1). This paper investigates an improved interpolation obtained by applying (usually) one Jacobi relaxation step to (10) using the original AMG interpolation as first approximation.

Although the above algebraic principle is formally very general, there is one important aspect to be addressed, namely, the aspect of smoothing: approximations of (12) merely involve F-equations (in our case just F-relaxations are required) and, as such, do not take smoothing into account (updating only a subset of variables has no real smoothing properties). Consequently, approaches which merely rely on algebraically approximating A_{FF}^{-1}, may be very robust but they may also be very inefficient.

AMG, as we understand it, also attempts to exploit the advantages of smoothing by a proper selection of the C/F-splitting. In fact, the real motivation of the splitting used in AMG is not just to formally obtain diagonal dominance (as mentioned above) but rather to coarsen such that, in particular, (full) Gauss-Seidel relaxation has reasonable smoothing properties *relative to the coarser levels*: roughly speaking, coarsening is essentially "in the direction of strong couplings" (e.g., see Figure 3b). Incidentally, this then *leads to* diagonally dominant matrices A_{FF}. Therefore, we usually use *complete C/F-relaxation steps* rather than mere F-relaxations. The importance of this is not motivated by the above algebraic arguments. However, ignoring the C-relaxation part normally leads to drastically reduced efficiencies (see Section 4). On the other hand, the fact that we *can* enforce convergence even without using C-relaxations, is another reason of AMG's robustness

which makes it applicable also in cases where smoothing is a serious problem (see Section 4.3).

3 The AMG Algorithm

The application of AMG to a given problem is a two part process. The first part, a fully automatic *setup phase*, consists of choosing the coarser levels and defining the transfer and coarse-grid operators. The second part, the *solution phase*, is straightforward and just uses the resulting components in order to perform normal multigrid cycling until a desired level of tolerance is reached (involving point relaxation for smoothing). This section outlines the algorithm used in the setup phase.

According to Section 2.1, only the coarsening process itself (C/F-splitting) and the interpolation, I_{FC}, have to be explicitely defined. (Restriction is always taken as the transpose of interpolation, although this is not necessarily the best for non-symmetric problems, see Section 2.2.) The approach used in this paper is an extension of the one described in [19] the essential aspects of which, for convenience, are briefly re-called in the next section. Here and in the following, we omit indices h and H which distinguish fine and coarse levels. All of the following has to be repeated recursively for each level.

3.1 Original AMG Interpolation

The original AMG interpolation is based on the concept of *strength of connectivity* between variables. We define

$$\mathbb{N}_i = \{j \in \Omega : j \neq i,\ a_{ij} \neq 0\}\ , \quad \mathbb{S}_i = \{j \in \mathbb{N}_i :\ -a_{ij} \geq \eta \max_{k \neq i, a_{ik} < 0} \mid a_{ik} \mid\}$$

and call $\mathbb{S}_i$ the set of *strong connections*[2] of the variable i, the default value of η being 0.25.

Although the construction of interpolation is strongly related to the coarsening process, for ease of description, let us assume a C/F-splitting of Ω to be given. The original AMG interpolation,

$$e_F = I_{FC}^{(0)} e_C \quad \text{or, explicitely,} \quad e_i = \sum_{k \in C} w_{ik}^{(0)} e_k \quad (i \in F)\ , \tag{14}$$

is obtained by approximating the equations (cf. (10))

$$a_{ii} e_i = -\sum_{j \in \mathbb{N}_i} a_{ij} e_j = -\sum_{j \in \mathbb{C}_i} a_{ij} e_j - \sum_{j \in \mathbb{D}_i^w} a_{ij} e_j - \sum_{j \in \mathbb{D}_i^s} a_{ij} e_j \quad (i \in F)\ . \tag{15}$$

We here subdivided the set $\mathbb{N}_i$ of all connections into three disjoint subsets defined by

$$\mathbb{C}_i = C \cap \mathbb{S}_i,\ \mathbb{D}_i = \mathbb{N}_i - \mathbb{C}_i,\ \mathbb{D}_i^w = \mathbb{D}_i - \mathbb{S}_i,\ \mathbb{D}_i^s = \mathbb{D}_i \cap \mathbb{S}_i\ .$$

[2] Note that, if A contains positive off-diagonal entries, the corresponding connections are not defined to be strong. Correspondingly, an F-variable i does not interpolate from any variable j with $a_{ij} > 0$. Unless such entries become substantial, AMG performance will not deteriorate essentially.

We now approximate (15) by first replacing e_j ($j \in \mathbb{D}_i^w$) with e_i, i.e. the corresponding off-diagonal entires are collapsed to the diagonal (weak couplings). For $j \in \mathbb{D}_i^s$ (strong couplings) this is generally not sufficient. Instead, we replace corresponding e_j's by the weighted average

$$e_j \longrightarrow \Big(\sum_{l \in \mathbb{C}_i \cap \mathbb{S}_j} a_{jl} e_l\Big) \Big/ \Big(\sum_{l \in \mathbb{C}_i \cap \mathbb{S}_j} a_{jl}\Big) \quad (j \in \mathbb{D}_i^s) \ . \tag{16}$$

Performing these replacements, we obtain the weights $w_{ik}^{(0)}$ of interpolation in (14). Note that interpolation is only from strong connections, i.e. $w_{ik}^{(0)} \neq 0$ only if $k \in \mathbb{C}_i$. Note also that this interpolation has rowsum 1 for the i-th variable, if the corresponding rowsum in A is 0.

Clearly, this definition makes sense only under certain requirements on the splitting. In fact, as already mentioned above, the selection of the splitting and the definition of the weights are closely related processes. Roughly, we have to require that C is selected such that, for each $i \in F$, $\mathbb{C}_i$ contains sufficiently many connections. Moreover, for (16) to be meaningful, we have to ensure that each variable $j \in \mathbb{D}_i^s$ has a sufficiently strong connection to the set $\mathbb{C}_i$. The more strongly connected j is to $\mathbb{C}_i$, the more accurate we can expect (16) to be. Our algorithm requires at least one such strong connection.

The concrete coarsening algorithm, described in detail in [19], is fairly involved and shall not be repeated here. The main objectives, however, can simply be summarized as follows (see also Figure 1):

Objective 1: For each $i \in F$ we request that each $j \in \mathbb{S}_i$ should either be in C (and therefore used for interpolation), or should be strongly connected to at least one point in $\mathbb{C}_i$ (i.e. $\mathbb{S}_j \cap \mathbb{C}_i \neq \emptyset$).

Objective 2: C should be a maximal subset of indices with the property that no two C-variables strongly depend on each other.

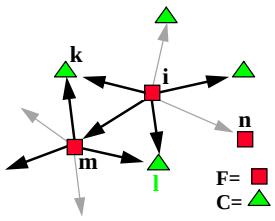

Figure 1: Illustration of the coarsening objectives (Objective 1). Black and grey arrows indicate strong and weak connections, respectively.

The first objective is the important one in order to obtain reasonable operator-dependent interpolation which, in turn, is necessary for fast and reliable AMG convergence. However, another major concern in the coarsening process is also to keep the final work per AMG cycle as low as possible. As a general rule, the larger C is, the better AMG convergence can be. On the other hand, the amount of work needed per cycle is directly related to the size of the coarse levels (and the resulting size of the Galerkin operators).

Therefore, the second objective has been introduced. Normally, it cannot strictly be satisfied. It rather serves as an additional guideline with the goal to limit the size of C while still enforcing the first objective.

Without going into detail, we want to note that the coarsening according to these objectives is realised by means of two successive algorithmic steps (this has turned out to be most efficient in practice). In the first step, AMG scans the set of unknowns and constructs a tentative splitting with the aim to merely satisfy Objective 2. In a second step, for each tentative F-variable, AMG checks whether or not the requests of Objective 1 are satisfied. In order to strictly satisfy Objective 1, some tentative F-variables may change their status and become C-variables in the final C/F-splitting. During this second step, also the interpolation weights are computed.

3.2 Improved Operator Dependent Interpolation

Since the tendency of the above coarsening strategy is to put a substantial part of strong connections into C, we can largely expect A_{FF} to be strongly diagonally dominant[3]. Therefore, a very simple way to improve interpolation is by applying Jacobi's relaxation to (10) using (14) as the first guess. More specifically, assuming any interpolation formula $e_F = I_{FC}^{(\mu)} e_C$ to be given, we derive an improved one, $e_F = I_{FC}^{(\mu+1)} e_C$, by defining

$$I_{FC}^{(\mu+1)} = (\mathbb{I}_{FF} - D_{FF}^{-1} A_{FF})\, I_{FC}^{(\mu)} \; - \; D_{FF}^{-1} A_{FC} \quad \text{with} \quad D_{FF} = diag(A_{FF})\,. \tag{17}$$

The explicit computation of this "relaxed interpolation" is fairly simple. However, each iteration introduces, roughly, a "new layer" of additional C-variables to be used for interpolation. Consequently, even if only one Jacobi step is applied on each AMG level, the resulting Galerkin operators will get less and less local towards coarser levels. However, most of the entries in the Galerkin operators are irrelevant for obtaining good convergence. Reasonable "truncation processes" are being developed, applied either directly to (10) or the resulting interpolation operator itself *before* the Galerkin operators are actually assembled. This work is not yet finished. Currently, the number of irrelevant Galerkin entries is reduced by just collapsing all entries to the diagonal which are smaller than 10^{-6} (relative to the diagonal). This has been done for ease of programming but it is a relatively costly process, too many irrelevant entries are still kept in the coarsening process and the average bandwidth of the Galerkin operators usually still grows towards coarser levels (which will become visible in the results presented in Section 4).

4 Results

In this section we present various experimental results demonstrating the effect of relaxation of interpolation on the convergence of AMG. For demonstration, we apply AMG to 2D model PDEs some of which can be regarded as "difficult" for standard multigrid

[3]Note that strong diagonal dominance can always be *enforced* in a controled way by just modifying the coarsening strategy slightly. As a potential consequence, more variables might be put into C thus increasing the overall complexity to a certain extent. This has not been done for the tests performed in this paper.

procedures. All problems are defined on the unit square with Dirichlet boundary conditions and AMG results are given for $n = N^2$ unknowns with N being the number of *inner* grid points in each direction. We want to recall, however, that AMG operates only on the given matrix problem (1), nothing else about the specific problems is known to AMG. (Note that we usually select *even* values for N which is quite inconvenient for geometric multigrid approaches since it corresponds to an *odd* number of subdivisions.)

Our main focus is on V-cycle convergence factors (more costly F-cycles are used only exceptionally) computed by a v. Mises iteration employing (at least) 70 multigrid cycles. Coarsening is always down to the coarsest possible level (containing a minimum of 9 variables) where the corresponding equations are solved directly. In order to give some indication on how computational work increases due to the more complex interpolation, the number of floating point operations per cycle is given.

We will consider various types of V-cycles denoted by "$V(\mu-\text{relax})$", where μ characterizes the interpolation and "relax" describes the smoothing steps (always applied before *and* after coarse-grid correction steps). For instance, V(0-CF) means that the original AMG interpolation (14) is used and smoothing is in C-F order (first the C-variables are relaxed by plain Gauss-Seidel and then the F-variables). Analogously, V(1-CFF) means that the original interpolation has been modified by one Jacobi step and smoothing is CFF (Gauss-Seidel applied once to the C-variables and then twice to the F-variables).

According to the algebraic motivation given in Section 2.3, mere F-relaxation steps should suffice to obtain convergence if interpolation is a sufficiently good approximation to (10). Therefore, below, we will specifically consider cycles employing merely F-relaxations such as V(0-FF) or V(1-FF). However, it turns out that such cycles, normally, converge poorly compared to those involving complete CF-relaxation steps (for a discussion, see Section 2.3). On the average, the V(1-CFF) cycle turned out to be the best choice in terms of robustness and convergence (not necessarily in terms of computational work, though). Here, the CF-part of relaxation makes a contribution to the smoothing, while the additional F-relaxation algebraically forces the range of relaxation to be closer to the range of interpolation. It turned out that, in all cases tested, further C- or F-steps do not pay. In particular, V(1-CFCF)-cycles are no better than V(1-CFF)-cycles.

4.1 Poisson-like Problems

For (isotropic) Poisson-like problems, the original AMG is known to work very efficiently [19] and there is no reason to expect relaxation of interpolation to have a substantial advantage over the original AMG interpolation (see, however, Section 4.5.1). This is demonstrated below for the 5-point Poisson discretization with Dirichlet boundary conditions on the unit square.

Figure 2a demonstrates the fast convergence of the V(0-CF)-cycle for grid sizes $N = 32, ..., 512$. It also clearly shows that the convergence of the V(0-FF)-cycle is far from being satisfactory: its asymptotic convergence factor for fine grids exceeds 0.7. Relative to this, the V(1-FF)-cycle converges considerably faster since the relaxed interpolation provides a better approximation to (10), see Figure 2b. This is due to the strong diagonal dominance of the submatrices A_{FF} on all coarser levels. However, within the range of N considered here, the V(1-FF)-cycle convergence is h-dependent and still much slower (and, for $N = 512$, approximately 1.6 times more expensive) than the V(0-CF)-cycle.

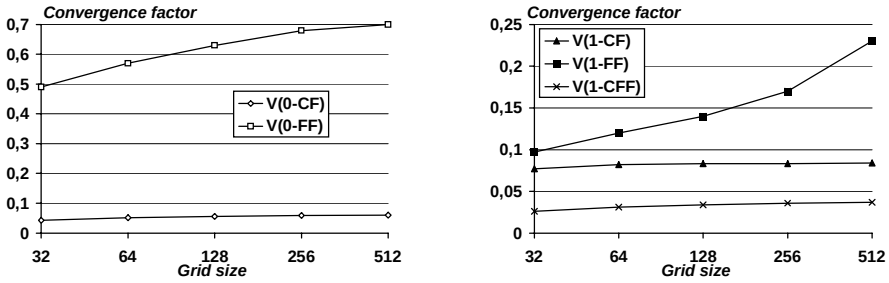

Figure 2: 5-point Poisson operator: convergence factors for cycles a) without and b) with relaxation of interpolation

On the other hand, the V(1-CF)-cycle is more expensive than the V(0-CF)-cycle by a factor of approximately 1.5 but it is not faster, see Figure 2b. Thus, relaxation of interpolation does not pay here. Note, however, that V(1-CFF) converges somewhat faster than V(1-CF): the additional F-relaxation, which enforces the ranges of interpolation and the smoothing operator to be closer together, has a visible effect. We point out that additional C- or F-relaxations do not improve convergence further. In particular, V(1-CFCF) is no faster than V(1-CFF).

Summarising, cycles employing merely F-relaxations cannot compete here with those involving complete CF-relaxation steps the reason being that, in contrast to CF-relaxations, mere F-relaxations have no smoothing properties. Getting satisfactory convergence with mere F-relaxations requires considerably more work in the sense of approximating (10). This has to be expected also for any other Poisson-like problem.

4.2 Interface Problems

Interface problems

$$-(D_1\, u_x)_x \;-\; (D_2\, u_y)_y \;=\; f(x,y) \tag{18}$$

with discontinuous coefficients $D_1 > 0$ and $D_2 > 0$ will, generally, not provide particular difficulties for AMG, although the convergence may depend somewhat on the concrete situation. In fact, this type of problem belonged to the target of AMG when it was originally developed.

We here consider a case which, according to [13], can be regarded as difficult even for robust geometric multigrid methods: the domain is the unit square and $f(x,y)$ is defined to be 0 except for the points $(0.25, 0.25)$, $(0.5, 0.5)$ and $(0.75, 0.75)$ where it is defined to be 10. Dirichlet boundary conditions are given as

$$u = 1 \quad \text{for} \quad x \le 0.5,\, y = 0 \text{ and } x = 0,\, y \le 0.5\,; \quad \text{otherwise}: \; u = 0\,.$$

The distribution of the discontinuous coefficients is indicated in Figure 3a. In contrast to [13], we discretize (18) in the usual way rather than using the harmonic average to compute diffusion coefficients between vertices (which tends to smear the discontinuities).

Figure 3b depicts how AMG adapts its coarsening (first 5 levels) to the particular requirements of the problem at hand. The figure shows all grid points of the finest level,

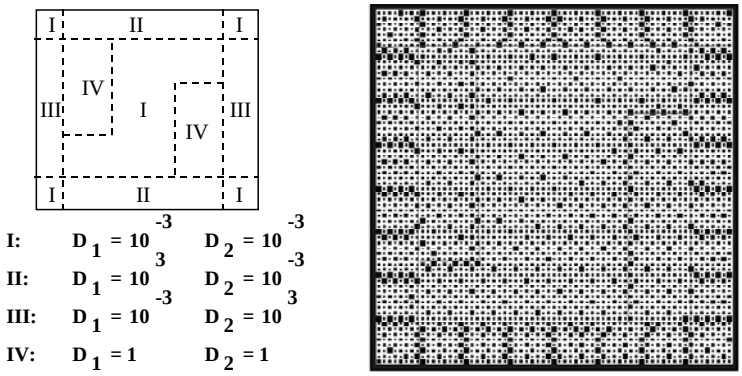

Figure 3: Interface problem: a) distribution of coefficients b) AMG coarsening

highlighting those which stay also on the coarser levels (the fatter a point, the longer it stays in the coarsening process). One realises that AMG locally performs correct semi-coarsening where necessary (subdomains II and III) and fairly regular coarsening otherwise. Moreover, close to the discontinuities, coarsening is somewhat slower than elsewhere.

It is this flexibility in adapting the coarsening, together with proper operator dependent interpolation, which makes AMG highly suitable for this kind of problem. In fact, Figure 4a shows that the V(0-CF)-cycle converges fast and, for fine meshes, convergence becomes essentially mesh independent. The observed convergence factor of 0.2 is typical for problems of this kind. In contrast to Poisson-like problems, however, relaxation of interpolation here leads to a substantial convergence improvement: in particular, the V(1-CFF)-cycle is more than twice as fast as the V(0-CF)-cycle for fine grids and convergence factors are virtually constant as a function of N. However, it is also twice as expensive (see Figure 4b) and, therefore, the extra effort does finally not pay. On the other hand, as before, cycles employing merely F-relaxations are much less efficient. In particular, for fine meshes, the V(1-FF)-cycle approaches the convergence of the V(0-CF)-cycle but it is approximately more expensive by a factor of 1.8. Moreover, the difference in convergence between the V(1-FF) and the V(1-CFF)-cycle clearly indicates that the additional C-relaxation step is very effective and the exclusive use of F-relaxations is not recommendable.

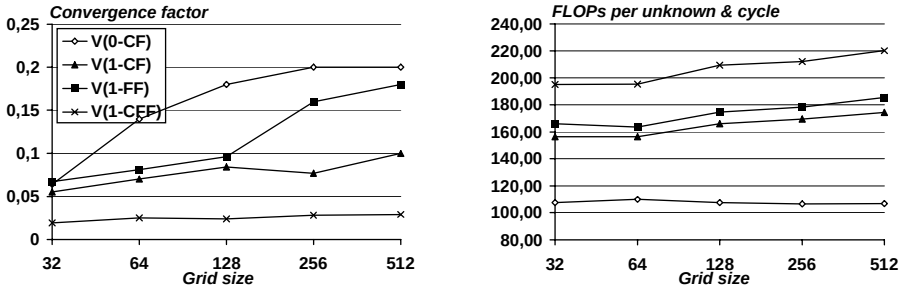

Figure 4: Interface problem: a) convergence factors b) computational work

We point out that the work (in terms of FLOPs per unknown and cycle) is slowly

increasing with N for all cycles involving relaxed interpolation. Concerning the reasons, we refer to the corresponding remark at the end of Section 3.2.

4.3 Rotated Anisotropic Diffusion Equation

An example which still today is frequently used for checking the robustness of multigrid methods, is the rotated anisotropic diffusion equation [23]

$$-(c^2+\varepsilon s^2)u_{xx} - 2(1-\varepsilon)scu_{xy} - (s^2+\varepsilon c^2)u_{yy} = f(x,y) \tag{19}$$

with $s = \sin\alpha$ and $c = \cos\alpha$. We consider this differential operator on the unit square with Dirichlet boundary conditions, $\varepsilon = 10^{-3}$ and various values of α. u_{xy} is discretized either by the left-oriented 7-point stencil (assuming $-90° \leq \alpha \leq 0°$) or by the central 4-point stencil, i.e.,

$$\frac{1}{2h^2}\begin{bmatrix} -1 & 1 & \\ 1 & -2 & 1 \\ & 1 & -1 \end{bmatrix} \quad \text{or} \quad \frac{1}{4h^2}\begin{bmatrix} -1 & & 1 \\ & 0 & \\ 1 & & -1 \end{bmatrix}. \tag{20}$$

The main difficulty with (19) is that it corresponds to the anisotropic operator $-u_{ss} - \varepsilon u_{tt}$ in a coordinate system obtained by rotating the (x,y)-system by an angle of α. Usual multigrid methods have serious difficulties because of the strong anisotropy which is not aligned with the grid. Consequently, neither point- nor line-relaxation schemes have good smoothing properties with respect to standard grid coarsening. Moreover, the extent to which the anisotropy is captured by grid points is different on different grid levels which also reduces the effectiveness of the coarse-grid correction process. Robust geometric multigrid approaches – involving usual operator dependent interpolation and Galerkin coarsening – still converge, but even if *F-cycles* are used as preconditioner for a conjugate gradient method, convergence may still be very slow (depending on α). This is particularly true if u_{xy} is discretized via the 4-point stencil (see, e.g., [13]).

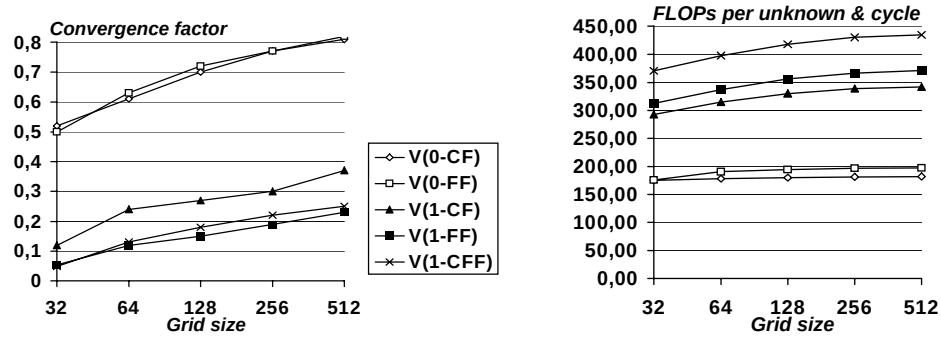

Figure 5: Rotated problem (7-point, $\alpha = -22.5°$): a) convergence b) work

This difficulty is also reflected in AMG but, due to the high flexibility in adapting the coarser levels, in a less severe manner. Even the original AMG, used with its default parameters, converges for any angle. Depending on the size of α, however, convergence becomes fairly slow. (Note that, except for very particular values of α, the resulting discretization matrices are not M-matrices.)

Figure 5a shows detailed results for the 7-point stencil and $\alpha = -22.5°$. Compared to the previously discussed problems, there are three major differences here. First, all cycles depicted in Figure 5a exhibit *h-dependent* convergence within the range of grids considered. Second, C-relaxation is *not* effective here. Obviously, smoothing is insufficient and convergence is mainly due to algebraic reasons as motivated in Section 2.3: the V(0-FF)-cycle behaves similar to the V(0-CF)-cycle and the V(1-FF)-cycle converges considerably faster than the V(1-CF)-cycle (it is even slightly faster than the V(1-CFF)-cycle). Finally, relaxation of interpolation gives a substantial improvement in cycle performance: while, for $N = 512$, both the V(0-CF) and the V(0-FF)-cycle converge at the fairly slow rate of 0.81, the V(1-FF)-cycle converges at a rate of 0.23. That is, the V(1-FF)-cycle is about 7 times faster than the V(0-CF)-cycle but only approximately twice as expensive (cf. Figure 5b).

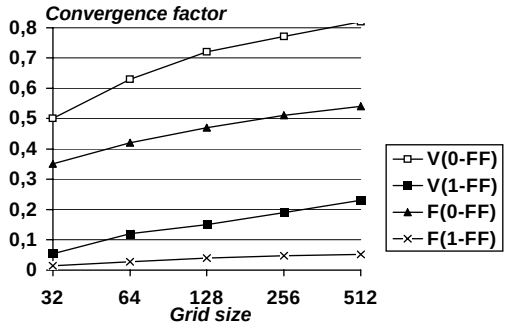

Figure 6: Rotated problem (7-point, $\alpha = -22.5°$): V-cycle vs. F-cycle

For the same case, Figure 6 presents a comparison between V- and F-cycle convergence. The figure shows that F-cycles converge substantially faster than their V-cycle analogs. In particular, the F(1-FF)-cycle converges at the very fast rate of 0.052.

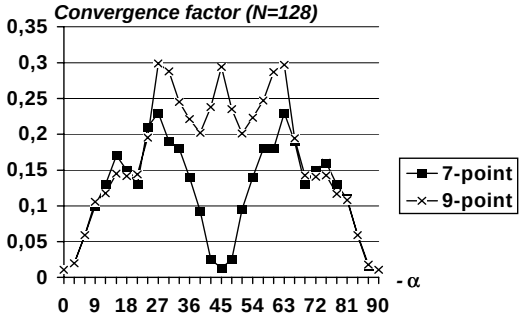

Figure 7: V(1-CFF)-convergence: 7-point versus 9-point discretization

For $N = 128$, Figure 7 gives an impression on the V(1-CFF)-cycle convergence as a function of α for both the 7- and 9-point discretizations. The figure shows that, for the 7-point case, AMG performs exceptionally well not only for α close to $0°$ and $-90°$ (in which case the anisotropy is just aligned with the axes) but also for $-45°$. In this case, AMG can still perfectly cope with the anisotropy since it is aligned with the diagonal of the grid, the corresponding stencil has only non-positive off-diagonal entries and essentially

degenerates to a 3-point stencil along the diagonal (left stencil):

$$\frac{1}{h^2}\begin{bmatrix} -\frac{1-\varepsilon}{2} & -\varepsilon & \\ -\varepsilon & 1+3\varepsilon & -\varepsilon \\ & -\varepsilon & -\frac{1-\varepsilon}{2} \end{bmatrix} \qquad \frac{1}{2h^2}\begin{bmatrix} -\frac{1-\varepsilon}{2} & -(1+\varepsilon) & \frac{1-\varepsilon}{2} \\ -(1+\varepsilon) & 4(1+\varepsilon) & -(1+\varepsilon) \\ \frac{1-\varepsilon}{2} & -(1+\varepsilon) & -\frac{1-\varepsilon}{2} \end{bmatrix} . \tag{21}$$

For the 9-point discretization, $\alpha = -45°$ does not play such a particular role (see right stencil in (21)) and AMG convergence is comparable to the worst 7-point cases. Apart from this particular difference, AMG performs very similarly for both discretizations. Concerning the convergence behavior of geometric multigrid processes, see [13].

4.4 Convection-diffusion Equation

Convection-dominated convection-diffusion problems also give rise to particular difficulties for standard multigrid methods. As an example, we here consider

$$-\varepsilon\,\Delta u \;+\; a(x,y)\,u_x \;+\; b(x,y)\,u_y \;=\; f(x,y) \tag{22}$$

with coefficients

$$a(x,y) = -\sin(\pi x)\cos(\pi y) \quad \text{and} \quad b(x,y) = \sin(\pi y)\cos(\pi x) \tag{23}$$

and $f(x,y) \equiv 1$ and $u = \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y)$ on the boundary of the unit square. We assume $\varepsilon = 10^{-5}$ and the first derivatives to be discretized by standard first order upwind differences. Note that the resulting discretization matrix is an M-matrix but not symmetric.

According to the results shown in [13], geometric multigrid approaches have serious difficulties with this example: covergence may be very slow and mesh dependent. One difficulty with this particular example is that a and b are chosen to yield *closed characteristics* and a stagnation point in the center of the domain. Consequently, (22) becomes more and more singular for $\varepsilon \longrightarrow 0$. For $\varepsilon = 0$, the continuous problem is no longer well defined: any function which is constant along the characteristic curves, solves the homogeneous equation.

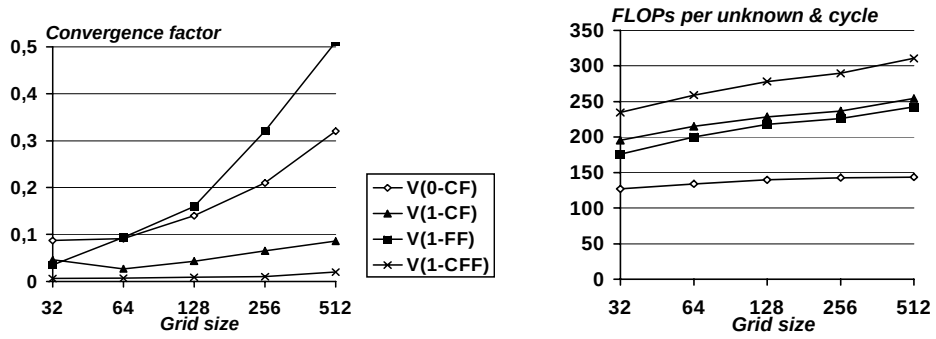

Figure 8: Convection diffusion equation: a) convergence b) computational work

Figure 8a shows the behavior of AMG. Using original AMG interpolation, the convergence is still fairly reasonable (see the V(0-CF)-cycle in the figure), at least within the

range of grids considered here. However, we can also observe a significant dependence of the convergence on the grid size. More smoothing does not help. On the other hand, relaxation of interpolation improves things substantially: the maximum asymptotic convergence factors of the V(1-CF) and V(1-CFF)-cycles are 0.086 and 0.020, respectively, and are largely mesh independent within the range considered here. In contrast to the rotated anisotropic case of the previous section, C-relaxation steps *are* very effective. In particular, the V(1-FF)-cycle convergence is strongly h-dependent and much slower than the V(1-CF) and V(1-CFF)-cycles.

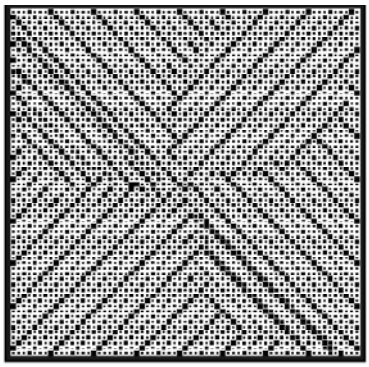

Figure 9: AMG coarsening for the convection diffusion problem

Figure 9 depicts the coarsening strategy performed by AMG. As mentioned earlier, the tendency of AMG is to coarsen "in the direction of strong couplings" based on the matrix entries. This is why AMG tries its best to not coarsen in the radial direction (within the limits imposed by the grid and the matrix entries).

4.5 Other Aspects

4.5.1 Accelerated Coarsening Strategies

Generally, the overall complexity of AMG directly depends on various aspects, one of which is the speed of coarsening. In order to allow for an "accelerated" coarsening, Objective 1 of Section 3.1 has to be weakened. Although such strategies are not a major topic of this paper, we want to give preliminary results for the 9-point Poisson discretization,

$$\frac{1}{h^2}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} . \tag{24}$$

A simple algorithm for "accelerated coarsening", which is still based on *direct* strong connections, can be constructed based on the following two objectives:

Objective 1: For each $i \in F$ we request that there is a strong connection to (at least) one C-variable.

Objective 2: C should be as small a set as possible subject to the previous objective.

Clearly, these objectives are much weaker than those given in Section 3.1. For 9-point stencils with only strong connections, for instance, they allow for an $h \to 3h$ coarsening (approximate reduction of the number of grid points by a factor of 9). Interpolation as described in Section 3.1 makes no sense anymore and is, just for the purpose of this section, replaced by the following one:

$$e_i = \sum_{k \in C} w_{ik}^{(0)} e_k \quad (i \in F) \quad \text{with} \quad w_{ik}^{(0)} := -a_{ik} / \sum_{j \in F} a_{ij} \,, \tag{25}$$

which, in terms of approximating (10), means that all off-diagonal entries in A_{FF} are simply collapsed to the diagonal. Geometrically speaking, this interpolation is rather crude. In particular, it allows for one-sided interpolation which may lead to insufficient, h-dependent AMG convergence (see the discussion in [19]). Applying AMG directly with this interpolation, cannot lead to an efficient method, not even for Poisson-like problems. For the 9-point stencil (24) this is confirmed by the results in Figure 10a which shows h-dependent V(0-CF)-convergence factors of over 0.95 for $N = 512$.

Note that this situation is very similar to the one arising for algebraic multigrid methods based on "aggregation of elements" (see, e.g., [3], [21], [6]). For this approach, one major problem is to find a good interpolation to begin with. The most natural one, "piecewise constant" interpolation has similar deficiencies as described above. To improve the resulting convergence, two approaches are followed: in [3], AMG is used as pre-conditioner in a conjugate gradient environment and in [21], [6], Jacobi relaxation is used to improve interpolation. In contrast to our "nested" approach, the aggregation approach requires the application of Jacobi relaxation to *all* equations.

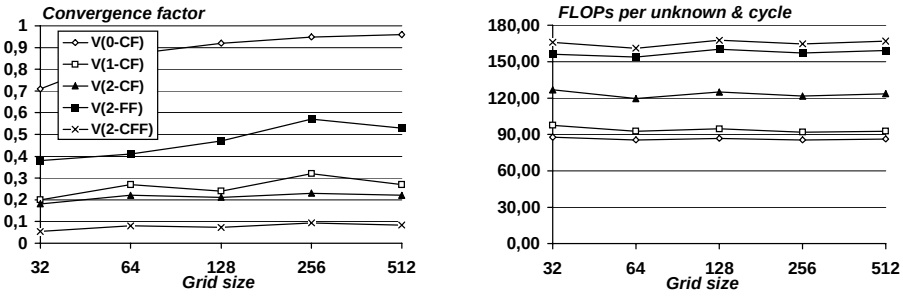

Figure 10: Accelerated coarsening: a) convergence factors b) computational work

Figure 10a shows that relaxation of interpolation drastically improves convergence. In particular, the asymptotic convergence of the V(2-CFF)-cycle is around 0.08. Note that, due to the rapid coarsening, there is only a relatively slow increase of numerical work with increasing μ (see Figure 10b).

Although accelerated AMG coarsening will naturally result in slower convergence and the overall computational time to solve a problem up to a fixed accuracy may finally not be reduced, it is still very important for practical applications: it is not just computational time but also storage which has to be kept low. The original AMG coarsening typically leads to complexities (= ratio of total storage and storage required for the given problem) of around 3 or even more. A substantial reduction of storage would often be very relevant, even if the overall computational work would not change essentially.

4.5.2 Aspects on Parallelisation

By partitioning the set of unknowns on the finest level, the *solution phase* of AMG can be parallelized in much the same way as any other multigrid method. Theoretically this is straightforward since only pointwise relaxation is performed for smoothing. However, due to the general and complex structure of AMG and due to non-regular sparsity patterns, in practice, parallelisation is technically rather complicated and involves very complex communications.

While some parts of the *preparation phase* can also be performed in parallel (like the evaluation of the Galerkin operators), the coarsening process itself is inherently sequential. As briefly sketched at the end of Section 3.1, coarsening is performed in two steps. Assuming a reasonable partitioning of the variables to be given for parallel processing, the first of these steps can formally be performed in parallel without any communication. The disadvantage of this is a potentially non-optimal distribution of variables along the inner interfaces (e.g. C-variables which are strongly connected across interfaces or too many neighboring F-variables along the two sides of an interface) which, in particular, may increase the final AMG complexity. The second step of the coarsening algorithm, the goal of which is to enforce Objective 1 (see Section 3.1), would normally require a lot of communication: processors would (in an unpredictable manner) need to change the status of variables (from F to C) which are located in different processors.

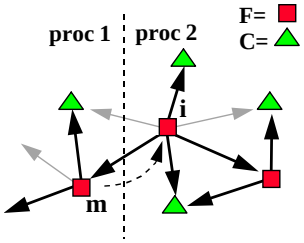

Figure 11: Modification of interpolation near processor boundaries (Method 1): Boldface and grey arrows denote strong and weak connections, respectively.

In order to avoid this, we try to satisfy Objective 1 as good as possible under the restriction that we do not allow a processor to change the status of any variable which is located in another processor. Near interfaces, at F-variables i for which the algorithm normally cannot satisfy Objective 1 in the best way without modifying the status of some neighboring F-variable, m, located in another processor (see Figure 11), we propose the following two different approaches.

Method 1: *Ignore variable m in interpolation.* That is, in defining interpolation for variable i, collapse the matrix entry a_{im} to the diagonal a_{ii} (cf. Figure 11). Due to the violation of Objective 1 for certain interface variables, the reduced accuracy of interpolation may have a more or less serious influence on the final convergence in the AMG solution phase. Here, a posteriori relaxation of interpolation (either global or just local to the boundary) can be used to largely eliminate possibly negative effects.

Method 2: *Make i itself a C-variable.* That is, Objective 1 is enforced in a very con-

servative way. There should be no substantial negative effects on convergence but, depending on the concrete situation, quite many additional C-variables might be introduced near interfaces which, in turn, may increase the total complexity of AMG substantially.

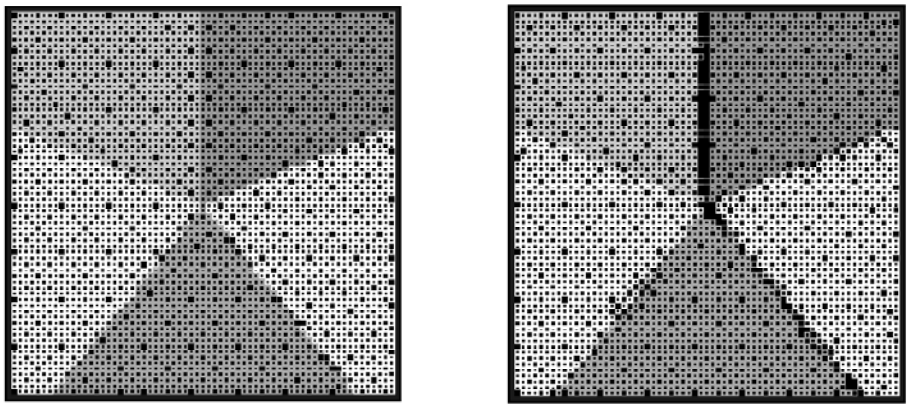

Figure 12: AMG parallel coarsening strategy: a) Method 1, b) Method 2

Our AMG program is not yet available in parallel. Instead, we have introduced the algorithmic modifications proposed above in our *serial program* in order to investigate the influence of a "parallel coarsening strategy" on the AMG convergence. Exemplary coarser levels resulting from applying Method 1 and Method 2, respectively, are depicted in Figure 12a-b for the case of Poisson's equation (assuming 5 processors). Figure 13a shows AMG convergence results for the convection-diffusion case (see Section 4.4), $N = 128$ and various processor configurations. The following methods are compared:

M1: Modified coarsening according to Method 1; V-cycles with CF-relaxation in the solution phase.

M1x: Modified coarsening according to Method 1; improvement of interpolation by applying one step of Jacobi's relaxation *local to the interfaces*; V-cycles with CF-relaxation in the solution phase.

M1xx: Modified coarsening according to Method 1; improvement of interpolation by applying one *global* Jacobi relaxation step; V-cycles with CFF-relaxation in the solution phase.

M2: Modified coarsening according to Method 2; V-cycles with CF-relaxation in the solution phase.

For M1, convergence strongly depends on the processor configuration. Improving interpolation by relaxation, either locally (M1x) or globally (M1xx), is very effective in eliminating this drawback, in particular, in achieving convergence which is largely independent of the processor configuration. In this example, local relaxation of interpolation is sufficient in order to maintain the convergence of the sequential V(0-CF)-cycle. Although

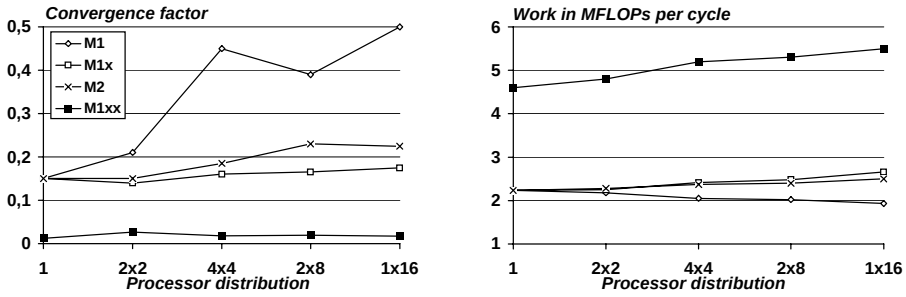

Figure 13: Parallel coarsening strategies: a) convergence b) computational work

for $N = 128$, as used here, the M1xx convergence is still considerably faster, its performance is comparable to M1x if numerical work is taken into account (cf. Figure 13b). (According to earlier results, depicted in Figure 8, this may change for finer grids.) The performance of M2 is comparable to that of M1x but M2 is much easier to realise since it does not require extra communication. In general, however, the additional C-variables near the interfaces may seriously disturb the coarsening pattern and, as a consequence, the overall complexity.

Although these results are very preliminary, they indicate that (local) relaxation of interpolation is one possibility to largely remove the disadvantages introduced by performing the AMG coarsening in parallel.

References

[1] Axelsson, O.; Vassilevski, P.S.: *Algebraic multilevel preconditioning methods I*, Num. Math. 56, 157-177, 1989.

[2] Axelsson, O.; Vassilevski, P.S.: *Algebraic multilevel preconditioning methods II*, SIAM Numer. Anal. 27, 1569-1590, 1990.

[3] Braess, D.: *Towards Algebraic Multigrid for Elliptic Problems of Second Order*, Computing 55, 379-393, 1995.

[4] Brandt, A.: *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, Arbeitspapiere der GMD 85, 1984.

[5] Brandt, A.: *Algebraic multigrid theory: the symmetric case*, Procs. of the International Multigrid Conference, Copper Mountain, Colorado, 1983.

[6] Chan, T.; Zikatanov, L.; Xu, J.: *Agglomeration strategy for unstructured grids*, AMLI'96, Proceedings of the Conference on "Algebraic Multilevel Iteration Methods with Applications" (Axelsson, O.; Polman, B. eds.), Nijmegen, June 13-15, 1996.

[7] Chang, Q.; Wong, Y.S,; Fu, H.: *On the algebraic multigrid method*, J. Comp. Phys. 125, 279-292, 1996.

[8] Dahmen, W.; Elsner, L.: *Algebraic multigrid methods and the Schur complement*, Notes on Numerical Fluid Mechanics 23, Vieweg Verlag, 1988.

[9] Fuhrmann, J.: *Outlines of a modular algebraic multilevel method*, ISSN 0946-8633, Weierstraß-Institut für Angewandte Analysis und Stochastik, Berlin, 1995.

[10] Grauschopf, T.; Griebel, M.; Regler, H.: *Additive multilevel-preconditioners based on bilinear interpolation, matrix dependent geometric coarsening and algebraic multigrid coarsening for second order elliptic PDEs*, TUM-19602, SFB-Bericht Nr. 342/02/96 A, Institut für Informatik, Technische Universität München, 1996.

[11] Griebel, M.; Neunhoeffer, T.; Regler, H.: *Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries*, SFB-Bericht Nr. 342/01/96 A, Institut für Informatik, Technische Universität München, 1996.

[12] Lonsdale, R.D.: *An algebraic multigrid solver for the Navier-Stokes equations on unstructured meshes*, Int. J. Num. Meth. Heat Fluid Flow 3, 3-14, 1993.

[13] Oosterlee, C.W.; Washio, T.: *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems*, to appear: SIAM SISC. Also available as: Arbeitspapiere der GMD, No. 980, 1996.

[14] Raw, M.: *A coupled algebraic multigrid method for the 3D Navier-Stokes equations*, Advanced Scientific Computing Ltd., 554 Parkside Drive, Waterloo, Ontario N2L 5Z4, Canada.

[15] Reusken, A.A.: *A multigrid method based on incomplete Gaussian elimination*, Eindhoven University of Technology, Report RANA 95-13, ISSN 0926-4507, 1995.

[16] Ritzdorf, H.; Stüben, K.: *Adaptive multigrid on distributed memory computers*, Multigrid Methods IV (Hemker, P.W.; Wesseling, P., eds.), Birkhäuser Verlag, p. 77, 1993.

[17] Robinson, G.: *Parallel computational fluid dynamics on unstructured meshes using algebraic multigrid*, Parallel Computational Fluid Dynamics 92 (Pelz, R.B.; Ecer, A.; Häuser, J. eds.), Elsevier Science Publishers B.V., 1993.

[18] Ruge, J.W.; Stüben, K.: *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, Multigrid Methods for Integral and Differential Equations (Paddon, D.J.; Holstein H.; eds.), The Institute of Mathematics and its Applications Conference Series, New Series Number 3, pp. 169-212, Clarenden Press, Oxford, 1985.

[19] Ruge, J.W.; Stüben, K.: *Algebraic Multigrid (AMG)*, In "Multigrid Methods" (McCormick, S.F., ed.), SIAM, Frontiers in Applied Mathematics, Vol 5, Philadelphia, 1986.

[20] Stüben, K.: *Algebraic multigrid (AMG): Experiences and comparisons*, Appl. Math. Comp. 13, pp. 419-452, 1983.

[21] Vanek, P.; Mandel, J.; Brezina, M.: *Algebraic multigrid on unstructured meshes*, University of Colorado at Denver, UCD/CCM Report No 34, 1994.

[22] Webster, R.: *An algebraic multigrid solver for Navier-Stokes problems in the discrete second order approximation*, Int. J. Num. Meth. in Fluids 22, 1103-1123, 1996.

[23] Wesseling, P.: *An Introduction to Multigrid Methods*, John Wiley, Chichester, 1992